Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2008 Proceedings

Americas Conference on Information Systems (AMCIS)

2008

An Ontology-based Approach for Model Representation, Sharing and Reuse

Amit V. Deokar
Dakota State University, Amit.Deokar@dsu.edu

Omar F. El-Gayar
Dakota State University, omar.el-gayar@dsu.edu

Nazim Taskin
Dakota State University, ntaskin@pluto.dsu.edu

Ruba Aljafari
Dakota State University, raljafari@unomaha.edu

An Ontology-based Approach for Model Representation, Sharing and Reuse

Amit V. Deokar
Dakota State University
Madison, SD 57042
amit.deokar@dsu.edu

Omar F. El-Gayar
Dakota State University
Madison, SD 57042
omar.el-gayar@dsu.edu

Nazim Taskin
Dakota State University
Madison, SD 57042
ntaskin@pluto.dsu.edu

Ruba Aljafari
Dakota State University
Madison, SD 57042
rhaljafari@pluto.dsu.edu

ABSTRACT

Although Decision Support Systems (DSS) play a dominant role in organizing data and models, its capability in supporting decision makers in collaborating distributed environments is still limited when it comes to the selection, sharing, and re-use of models. For mathematical models to be shared and reused, mechanisms are needed for understanding, implementing, modifying, discovering, selecting, engaging, and composing models. At a fundamental level, model representation will need to extend beyond model structure to include model semantics as well.

This research leverages advances in Semantic Web technologies and ontologies to enable sharing and re-using of decision models by providing enriched semantics in collaborative decision making environments. The proposed approach builds on structured modeling (SM) as an underlying modeling formalism and is illustrated using the Web Ontology Language (OWL). A case study demonstrates the viability of the approach for capturing model semantics models using ontologies.

Keywords

Decision support, model management, model representation, structured modeling, model sharing, ontology

INTRODUCTION

During the last few decades, behavioral and technical aspects of decision making in organizations have received considerable attention from researchers. With regard to tools, Decision Support Systems (DSS) that provide seamless access to relevant data and decision models alike represent a key element in supporting decision making. Similar to data, quantitative decision models are a critical organizational resources (Iyer, Shankaranarayanan and Lenard, 2005). However, recent advances in information technology infrastructure and the proliferation of distributed computing architectures impose technical challenges pertaining to information management tasks such as management of data, models and meta-models. Although DSS play a dominant role in organizing data and models, its capability in supporting decision makers in collaborating distributed environments is still limited when it comes to the selection, sharing, and re-use of models. Of particular relevance is the ability to represent models at a higher level of abstraction.

Seminal contributions by Blanning (1986), Geoffrion (1987), Dolk (1988) as well as recent contributions by Kim (2001), and El-Gayar and Tandekar (2007) represent a significant step towards a unified and standard representation facilitating model sharing and reuse. However, while such approaches capture model structure and allow for different types of models constructed and used in diverse modeling platforms, they do not explicitly capture model semantics. However, to further facilitate and automate (e.g., through the use of software agents) model sharing and reuse, it is paramount to capture model context and semantics (Chaari, Ejigu, Laforest and Scuturici, 2007). Mechanisms are needed for understanding, implementing, modifying, discovering, selecting, engaging, and composing models. In effect, model representation needs to extend beyond model structure to include model semantics as well.

In this research, we leverage advances in Semantic Web technologies and ontologies to propose model representation that captures model semantics and structure in a machine-understandable manner for computational processing of decision models, thus enabling their sharing and reuse in distributed settings. Our model representation approach is based on the structured modeling (SM) paradigm for management science models (Geoffrion, 1987). The proposed approach is illustrated utilizing the capabilities of the Web Ontology Language (OWL), which is a particular ontology representation language..

The remainder of the paper is organized as follows. The next section briefly describes related work about model representation, structured modeling, knowledge representation and ontologies. The subsequent section describes the proposed approach for model representation using ontologies and illustrates ontologies for representing model schemas and model instances utilizing the OWL. The following case study demonstrates the applicability of the proposed approach. Last but not least are concluding remarks and directions for future research.

RELATED WORK

Model Representation

The concept of Model Management (MM) evolved around the mid seventies (Sprague and Watson, 1975). MM as a function, is concerned with computer tools for processing quantitative decision models (Chang, Holsapple and Whinston, 1993). The objective is to leverage models as an organizational resource by effectively finding new ways for representing, developing, storing, controlling, and executing models (Krishnan and Chari, 2000). Research is moving towards adapting and reusing models, which represent organizational knowledge nuggets and can be potentially leveraged for effective and efficient knowledge exchange in the context of intra- and inter-organizational processes. Distributed collaborative model management is an evolving paradigm in the MM area that is attracting more research efforts since mid-1990s, and has become critical with penetration of global markets, mergers, and acquisitions (Deokar and El-Gayar, 2008). The ability of evaluating and understanding models in distributed work systems is critical along with provisioning of easy access and integration of models (Iyer et al., 2005).

Efforts in model representation are geared towards representing model at higher level of abstraction to facilitate model sharing and reuse (El-Gayar and Tandekar, 2007). For example, Bhargava (1993) suggested a logic-based approach for model representation of expressions. Other approaches focused on the use of the relational model in database domain (Blanning, 1986). Modeling languages like AMPL and LINGO are meant to facilitate modeling as well by using ad-hoc programs or matrix generator programs.

Among the aforementioned approaches, Geoffrion's (1987) Structured Modeling (SM) model representation approach is particularly attractive. Specifically, SM has many features that are highly desirable from a MM perspective (Krishnan and Chari, 2000), most notably are the independence of model representation and model solution, the sufficient generality to encompass a wide variety of modeling paradigms, and the representational independence of general model structure from the detailed data needed to describe specific model instances. In SM, a model is defined as a combination of a model schema and one or more model instances (separately from the model schema). A model schema describes the general structure of a model and is represented as a hierarchical, acyclic, and attributed graph. A model schema may be associated with one or more model instances. Model instances correspond to the data part of the model. SM has been shown to be particularly amenable to model management tasks such as model formulation, model composition, and model integration (Holocher, Michalski, Solte and Vicuna, 1997; Liang and Konsynski, 1993).

Based on SM and eXtensible Markup Language (XML) schema, Structured Modeling Markup Language (SMML) was developed by El-Gayar and Tandekar (2007) to represent, share, and re-use various models in different modeling environments. In a distributed environment, SMML is a significant extension to SM as models represented in SMML are serialized as XML files and thus amenable to sharing and reuse in a distributed environment. However, neither SM nor SMML are able to capture model semantics such as the meaning of various constructs, the additional details associated with these constructs (such as measurement units) and how these constructs relate to other concepts from the problem domain. Model semantics, if any, are only informally captured in the form of comments suitable for human consumption, rather than computational reasoning. The proposed framework addresses this major limitation by leveraging recent developments in the Semantic Web and ontologies to provide semantically rich representation of models.

Knowledge Representation and Ontologies

Decision models are knowledge objects that capture valuable organizational know-how at operational, tactical, and strategic levels. In order to create computer supported environments such as intelligent decision support systems that encapsulate these models to solve business problems, it is imperative that their representation schemes promote computational reasoning capabilities. In this respect, the notion of Knowledge Representation (KR) is relevant in the context of model representation. KR has a long historical background in Artificial Intelligence (AI) research, where the focus has been on structuring and encoding knowledge in different forms, used in conjunction with inference procedures, for the development of intelligent systems (Russell and Norvig, 2002). According to Davis, Shrobe and Szolovits (1993), KR plays multiple roles, in that it is a surrogate, a set of ontological commitments, a fragmented theory of intelligent reasoning, a medium for efficient computation, and a medium of human expression.

Ontologies provide a conceptualization mechanism or a vocabulary to represent knowledge in a given domain, and are sometimes referred to as "content theories" (Chandrasekaran, Josephson and Benjamins, 1999). The concept of ontologies has been widely studied and definitions from different perspectives abound. More recently, the notion of ontologies has been referred by the Semantic Web community members as "a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some particular topic" (Hendler, 2001). Another relevant definition provided by Studer, Benjamins and Fensel (1998) suggests ontologies as "a formal, explicit specification of a shared conceptualization." Researchers have also proposed various categorizations for ontologies. For instance, Guarino (1998) distinguishes between top-level ontologies, domain ontologies, task ontologies, and application ontologies.

It is important to note that the major goal of ontologies is not merely to serve as taxonomies or vocabularies, but that of knowledge sharing and reuse by applications and systems. Neches, Fikes, Finin, Gruber, Senator and Swartout (1991) point out many modes of knowledge sharing and reuse enabled by ontologies including exchange of techniques, inclusion of source specifications (at design-time), run-time invocation of external modules or services, and interoperability between systems through communication. These modes of sharing and reuse also apply in the context of model management.

Ontological engineering has grown as a subarea within knowledge engineering that concerns with ontology development and use throughout the ontology life cycle – design, implementation, validation, deployment, maintenance, mapping, sharing, and reuse (Gomez-Perez, Corcho and Fernandez-Lopez, 2004).

Ontology representation languages play a key role in ontological engineering by providing a means to build ontologies based on specific KR paradigms to formally represent different knowledge modeling components (such as concepts, and roles). Earlier ontology representation languages such as KIF (Genesereth and Fikes, 1992) and Ontolingua (Gruber, 1992) were based on KR techniques such as first-order logic and frame-based representation. Recently, XML-based ontology representation languages, also called as ontology markup languages, have emerged to support ontology representation in the context of the Semantic Web (Pulido, Ruiz, Herrera, Cabello, Legrand and Elliman, 2006). Resource Description Framework (RDF) is a W3C (World Wide Web Consortium) recommendation developed for describing Web resources with metadata and incorporates a data model based on the semantic-network KR paradigm (W3C, 2004). RDF Schema (RDFS) is an extension of RDF with frame-based primitives for defining the relationships between properties and resources, and is also a W3C Recommendation (W3C, 2004). Ontology Inference Layer (OIL), based on the Description Logics (DL) KR paradigm (Baader, Calvanese, McGuinness, Nardi and Patel-Schneider, 2003), is an extension of RDF/RDFS adding more frame-based representation primitives and eluding the RDF reification mechanism (Fensel, Horrocks, van Harmelen, Decker, Erdmann and Klein, 2000). DAML+OIL, also based on DL KR paradigm (Baader et al., 2003), is an evolution of the earlier DARPA Agent Markup Language (DAML) attempting to combine the expressiveness of DAML and OIL by providing DL extensions of RDF/RDFS directly (van Harmelen, Patel-Schneider and Horrocks, 2001). Web Ontology Language (OWL), derived from the DAML+OIL language, is a W3C recommendation and is the current standard ontology markup language for the Semantic Web (W3C, 2004). It is extremely rich for describing relationships among class, properties, and individuals. Its vocabulary includes support for property type restrictions, equality, property characteristics, class intersection, and restricted cardinality. Additionally, OWL is not a closed language; it is instead a combination of three sublanguages with increasing expressiveness, namely OWL-Lite, OWL-DL, and OWL-Full, to support varying needs of knowledge engineers.

MODEL REPRESENTATION USING ONTOLOGIES

Ontologies can be used to develop semantically rich models that can support intelligent reasoning and querying based on not only syntactic information, but also semantic information. These reasoning capabilities provide the necessary technological support needed to discover, interpret, compose, and execute models. Moreover, the use of ontologies facilitates the capture of model semantics that is independent of a particular tool or application. To better understand the role of ontologies, the following section describes various levels of abstraction for representing models. Next, we illustrate an ontology-based model representation using OWL for describing model semantics along with model structure.

Representing Models at Various Levels of Abstraction

Model representation can be viewed at three levels of abstraction as shown in Figure 1. Level 1 indicates the highest level of abstraction, where the goal of representation is to denote a particular modeling paradigm in terms of its fundamental constructs and relationships among them. The overall notion is similar to meta-modeling that gives information about the feasible structure of a particular model schema or instance. In this article, we have focused on using Structured Modeling (SM) as the model representation paradigm, and hence this level represents the conceptual representation of the SM technique. This conceptual representation may be captured for the model schema as well as the model instance. The model schema structure captures the concepts and their relationships for a structured model schema, while the model instance structure captures the concepts and their relationships for denoting the elemental details in a structured model instance. This representation incorporates constructs to annotate various model schema structure concepts and relations, which can be used to provide model semantics at the next level.

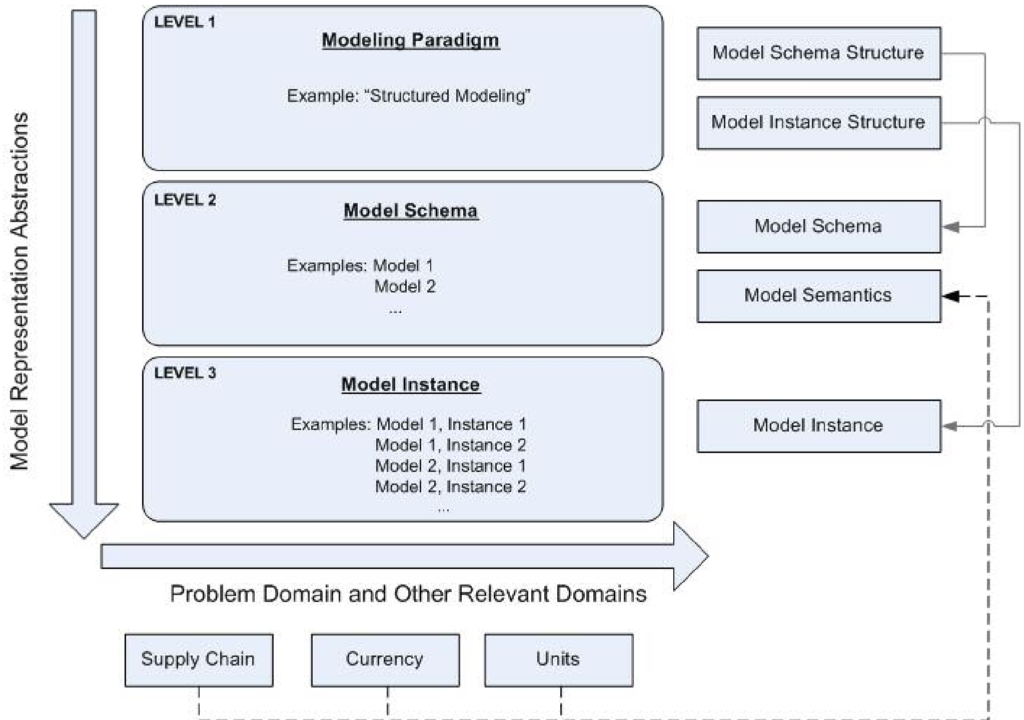

Figure 1. Understanding Model Representation

The next lower level of abstraction is Level 2, where the goal is to represent a particular model, independent of its data, such that various sets of data values may be used to instantiate this model at the lowest level of abstraction, i.e. Level 3. For example, an optimization model for a transportation problem in the supply chain domain may be represented in a data-independent manner at this level. This level is critical in that it provides both structural and semantic information regarding the model under consideration. This is achieved primarily through instantiation of the Level 1 constructs in the context of the problem. Most of the constructs provide structural details about the model. However, certain constructs use interpretations through key phrases and free form text for providing semantic information regarding the model components. These constructs are linked to specific domain ontologies to provide extensible use and sharing of models. For example, the units for particular variables (e.g., dollars for cost variables) may be noted and linked to appropriate domain ontology (e.g., currency) for use of the model by an user wishing to use another unit system (e.g., euros for cost variables). Thus both model structure and semantics is captured at this level for any model amenable to the use of SM.

Finally, Level 3 indicates the lowest level of abstraction, where the goal is to represent a particular model instance, associated with a given data set. For example, an optimization model for a transportation problem in the supply chain domain may be instantiated with a particular data set where the parameter values needed to solve the problem are clearly provided. Model solvers can then use this model instance to provide results for the transformation problem for the given data set. Thus, this level provides structural information regarding the problem instance, achieved through instantiation of Level 1 constructs relating to the model instance structure. The semantic information related to the data values can be traced from the Level 2 as mentioned earlier. For instance, the semantics associated with a value of 250 that is provided for a cost variable in Level 3 can be analyzed from Level 2, where the cost may be expressed in dollars and linked to currency domain ontology.

As shown in Figure 1, the different abstraction levels for model representation serve as one dimension. Along an orthogonal dimension are the different domains. A number of domains may also be relevant in the context of a particular problem domain. For example, a transportation model may be primarily formulated for the supply chain domain. The supply chain domain ontology may consist of key terms such as supplier, demand and customer. Other domain ontologies can provide additional semantics to the model structure. For example, a currency ontology as mentioned in the discussion above may be used to provide semantics to cost variables. Another units ontology may provide semantics associated with units such as tons. Thus, in a nutshell, the problem domain ontology along with other auxiliary domain ontologies can form a library of referring ontologies to provide semantics to the models.

Model Schema Structure Ontology (Level 1) using OWL

Figure 2 shows a model schema structure in a UML class diagram based on SMML (El-Gayar and Tandekar, 2007). The class diagram shows the key constructs in a model schema and their relationships in a graphical manner. The corresponding OWL ontology is also shown adjacent to it. In accordance with the SM notation (Geoffrion, 1987), the key constructs of a model schema are model, genus, and module. A model can be comprised of one or more genera or modules with their names, a genus type indicator (`pe` for primitive genera, `ce` for compound genera, and so on), index set, range (in case of attribute genera), a calling sequence (in case of non-primitive genera), and a rule (in case of function or test genera). Similarly, a module can be further comprised of one or more genera or modules with corresponding properties. Finally, a genus is a collection of all elements of a given type.

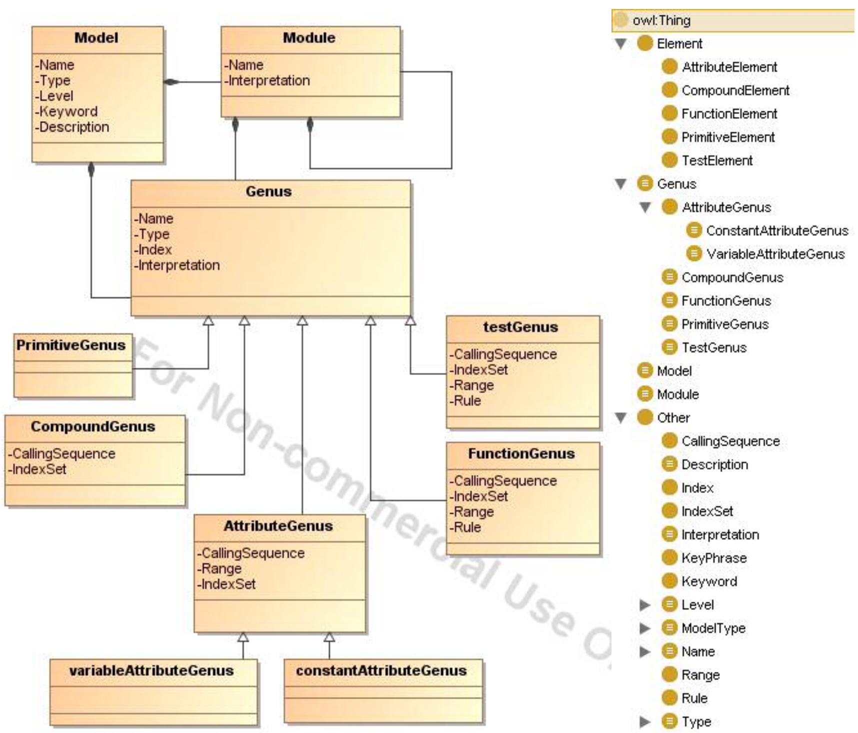

Figure 2. Model Schema Structure: UML and OWL Representations

Figure 3 shows a code snippet of the Genus OWL class. The `hasGenusType` property on the `Genus` class has a local range restriction associated with it, implying that the property is restricted to have `allValuesFrom` the class `Type`. This means that if an instance genus `CompoundGenus1` is related by the property `hasGenusType` to the instance `ce`, then from this a reasoned can deduce that `ce` is an instance of the class `Type`. However, the reasoner cannot deduce from an `allValuesFrom` restriction alone that there actually is at least one value for the property `hasGenusType`. Now, the `someValuesFrom` restriction on the `hasGenusType` property of the class `Genus` ensures that *some* value for the `hasGenusType` property should be an instance of the class `Type`. Thus, together the `allValuesFrom` and `someValuesFrom` properties constrain the `hasGenusType` property to define the class `Genus` appropriately. Other properties of the `Genus` class such `hasName`, `hasIndex`, `hasInterpretation`, and so forth are defined similarly. Also, other classes in the model structure schema such as `Model`, `Module`, `PrimitiveGenus`, and so forth also use similar closure axioms in their definitions. Note that the property `hasInterpretation` provides a mechanism to associate model semantics by linking key phrases to the relevant domain ontologies.

```
<owl:Class rdf:ID="Genus">
    <owl:equivalentClass>
        <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#hasGenusType"/>
                    <owl:allValuesFrom rdf:resource="#Type"/>
                </owl:Restriction>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#hasGenusType"/>
                    <owl:someValuesFrom rdf:resource="#Type"/>
                ...
            </owl:intersectionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasInterpretation"/>
            <owl:allValuesFrom rdf:resource="#Interpretation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasInterpretation"/>
            <owl:someValuesFrom rdf:resource="#Interpretation"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
```

Figure 3. Code Snippet of the OWL Class: Genus in the Model Schema Structure Ontology

Model Instance Structure Ontology (Level 1) Using OWL

Figure 4 shows a model instance structure in a UML class diagram based on SMML (El-Gayar and Tandekar, 2007). The class diagram shows the key constructs in a model instance and their relationships in a graphical manner. The corresponding OWL ontology is also shown adjacent to it. In accordance with the SM notation (Geoffrion, 1987), the key constructs of a model instance are elemental detail, parameter, table, field, record, and field value. The elemental detail is comprised of tables and parameters, and has a name, reference to a model schema, and a namespace. Each table consists of a number of fields and records. Each record includes any number of field values corresponding for the filed definition of that record. Finally, each parameter has a name, type, and a value.

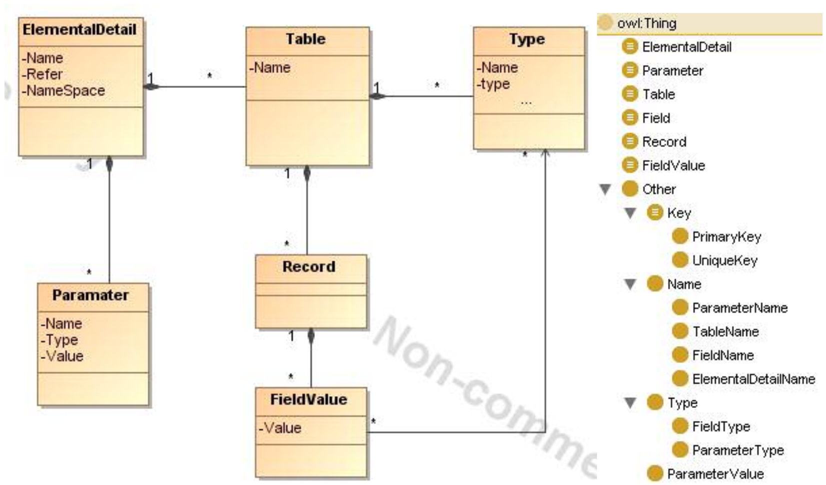

Figure 4. Model Instance Structure: UML and OWL Representations

CASE STUDY

A number of models representing the four levels of structured modeling that are defined by Geoffrion (1990; 1992a; 1992b) have been represented using OWL ontologies based on the model abstraction framework shown in Figure 1. For illustration purposes, this section describes the use of OWL ontologies for representing semantically rich models in the case of the well-known Hitchcock-Koopmans transportation example in the supply chain domain.

The transportation model is a linear programming problem used to compute the optimal annual flows. The main primitive elements of the transportation problem are *plants*, and *customers*. Every *plant* has a *supply capacity* measured in tons. Every *customer* has a nonnegative *demand* measured in tons. There are some transportation *links* from *plants* to *customers*. There must be at least one *link* incident to each *plant*, and at least one *link* incident to each *customer*. There can be nonnegative transportation *flow* (in tons) over each link. Every *link* has a *transportation cost rate* for use in $/ton. There is *a total cost* associated with all *flows*. Two tests are associated with this problem. The *supply tes*t determines if the total *flow* leaving a *plant* is less than or equal to its *supply capacity*. Similarly, the *demand test* determines if the total *flow* arriving at a *customer* is exactly equals to its *demand*. This is the essentially the summarized interpretation of the model. The italicized phrases are the key phrases in the interpretation.

The model schema ontology at abstraction level 2 (see Figure 1) is built by instantiating the model schema structure ontology discussed earlier (see Figure 2) based on the transportation model specifics. Similarly, the model instance ontology at abstraction level 3 (see Figure 1) is built by instantiating the model instance structure ontology discussed earlier (see Figure 4) based on the transportation model specifics.

Each of the genus and module model schema elements have the `hasInterpretation` property, which maps to any number of relevant key phrases. For example, "customer", "supply capacity", "demand" are key phrases. Each of these key phrases is mapped to specific domain ontologies. In this example, these phrases are mapped to a supply chain ontology. Models developed by different partnering organizations in a supply chain may share such different models, and still semantically connect to each other through these commonly shared domain ontologies.

Models using disparate, yet related terminology can be reasoned about by model management systems to perform functions like model searching, retrieval, composition, and so forth. Based on this extensible model representation, we are currently developing mechanisms to support these MM functionalities (Deokar and El-Gayar, 2008).

DISCUSSION AND CONCLUDING REMARKS

In this research, we leverage recent developments in Semantic Web technologies and ontologies to propose a richer model representation for supporting modeling sharing and reuse. Models may be represented using ontology representation languages (such as illustrated using OWL) by either developing such new model ontologies or by transforming existing ontologies represented in XML-based formats (such as SMML) and further enhancing them with semantic information from various domain ontologies (either newly developed or reused). The proposed approach is discussed in the context of structured modeling representation paradigm.

The main contribution of the proposed approach is the comprehensive ability to capture both structure and semantics of decision models. While existing approaches have successfully captured model structure at a higher level of abstraction, lack of model semantics has hampered leveraging these approaches for model sharing and reuse. The aforementioned limitation is further confounded with the greater emphasis on software agents and service oriented computing. The proposed approach addresses these limitations by explicitly capturing model semantics in tandem with model structure.

Model representation can be used in association in ontologies in multiple ways. An alternative approach would be to develop new or reuse existing domain ontologies to augment existing model representation formats such as SMML (El-Gayar and Tandekar, 2007). In essence, the role of domain ontologies becomes that of semantically annotating models represented in such XML-based formats. Here, the model structure is captured using the XML-based formats, while model semantics is captured using domain ontologies.

Still other approaches may be more suited for cases where model structure cannot made accessible explicitly, due to issues such as privacy and trust, especially in inter-organizational settings. Models may be available as executable software components or services (e.g., Web services), where explicit model representation details such as the discussed in above sections may not be available. In such cases, ontologies can be leveraged to annotate such models and provide details at varying levels (depending on access control mechanisms in place to respect privacy and security issues). Ontology markup languages such as OWL-S can present ontological details in a selective manner leveraging underlying detailed ontological descriptions (such as those discussed in this article in OWL), so as to provide the necessary machine-understandable semantics for intelligent reasoning and supporting computational model management functionalities.

Further research is warranted in the context of the above mentioned alternative approaches. Also, following avenues are viable future opportunities for research in this area: (1) Exploring the viability of the proposed approach on a comprehensive set of models spanning multiple modeling paradigms such as simulation, simultaneous differential equations, and stochastic models, (2) Exploring the applicability of other ontology languages, and (3) Leveraging semantically described models for various model management functions including but not limited to querying, enacting, reasoning, and composing models.

REFERENCES

1. Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. (Eds.) (2003) *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press, Cambridge, MA.
2. Bhargava, H. K., and Kimbrough, S. O. (1993) Model management - An embedded languages approach, *Decision Support Systems*, 10, 3, 277-299.
3. Blanning, R. W. (1986) An entity-relationship approach to model management, *Decision Support Systems*, 2, 1, 65-72.
4. Chaari, T., Ejigu, D., Laforest, F., and Scuturici, V. (2007) A comprehensive approach to model and use context for adapting applications in pervesive environments, *Journal of Systems and Software*, 80, 1973-1992.
5. Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R. (1999) What are ontologies and why do we need them?, *IEEE Intelligent Systems*, 14, 1, 20-26.
6. Chang, A.-M., Holsapple, C. W., and Whinston, A. B. (1993) Model management issues and directions, *Decision Support Systems*, 9, 1, 19-37.
7. Davis, R., Shrobe, H., and Szolovits, P. (1993) What is knowledge representation?, *AI Magazine*, 14, 1, 17-33.
8. Deokar, A. V., and El-Gayar, O. F. (2008) A semantic web services-based architecture for model management, in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS-41 '08)*, Waikoloa, Big Island, HI, IEEE Computer Society.
9. Dolk , D. R. (1988) Model management and structured modeling: the role of an information resource dictionary system,, *Communications of the ACM*, 31, 6, 704-718.
10. El-Gayar, O. F., and Tandekar, K. (2007) An XML-based schema definition for model sharing and reuse in a distributed environment, *Decision Support Systems*, 43, 791-808.
11. Fensel, D., Horrocks, I., van Harmelen, F., Decker, S., Erdmann, M., and Klein, M. (2000) OIL in a nutshell, in R. Dieng (Ed.) *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW '00)*, 1-16.
12. Genesereth, M. R., and Fikes, R. E. (1992) Knowledge Interchange Format, Version 3.0 Reference Manual, Technical Report Logic-92-1, Computer Science Department, Stanford University, Stanford, CA.
13. Geoffrion, A. M. (1987) An introduction to structural modeling, *Management Science*, 33, 5, 547-588.
14. Geoffrion, A. M. (1990) A library of structured models, *Informal Note*.
15. Geoffrion, A. M. (1992a) The SML language for structured modeling: Level-1 and Level-2, *Operations Research*, 40, 1, 38-57.
16. Geoffrion, A. M. (1992b) The SML language for structured modeling: Level-3 and Level-4, *Operations Research*, 40, 1, 58-75.
17. Gomez-Perez, A., Corcho, O., and Fernandez-Lopez, M. (2004) Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web, Springer.
18. Gruber, T. R. (1992) Ontolingua: A mechanism to support portable ontologies, Technical Report KSL-91-66, Knowledge Systems Laboratory, Stanford University, Stanford, CA.
19. Guarino, N. (1998) Formal ontology in information systems, in N. Guarino (Ed.) *Proceedings of Formal Ontology in Information Systems (FOIS '98)*, June 6-8, Trento, Italy, IOS Press, 3-15.
20. Hendler, J. (2001) Agents and the semantic Web, *IEEE Intelligent Systems*, 16, 2, 30-37.
21. Holocher, M., Michalski, R., Solte, D., and Vicuna, F. (1997) MIDA: An open systems architecture for model-oriened integration of data and algorithms, *Decision Support Systems*, 20, 2, 135-147.
22. Iyer, B., Shankaranarayanan, G., and Lenard, M. L. (2005) Model management decision environment: a Web service prototype for spreadsheet models, *Decision Support Systems*, 40, 2, 283-304.
23. Kim, H. (2001) An XML-based modeling language for the open interchange of decision models, *Decision Support Systems*, 31, 4, 429-441.
24. Krishnan, R., and Chari, K. (2000) Model management: Survey, future research directions and a bibliography, *Interactive Transactions of OR/MS*, 3, 1.
25. Liang, T.-P., and Konsynski, B. R. (1993) Modeling by analogy: Use of analogical reasoning in model management systems, *Decision Support Systems*, 9, 1, 113-125.
26. Neches, R., Fikes, R. E., Finin, T., Gruber, T. R., Senator, T., and Swartout, W. R. (1991) Enabling technology for knowledge sharing, *AI Magazine*, 12, 3, 36-56.

27. Pulido, J. R. G., Ruiz, M. A. G., Herrera, R., Cabello, E., Legrand, S., and Elliman, D. (2006) Ontology languages for the semantic web: A never completely updated review, *Knowledge-Based Systems*, 19, 489-497.
28. Russell, S., and Norvig, P. (2002) Artificial Intelligence: A Modern Approach, Second ed., Prentice Hall.
29. Sprague, R. H., and Watson, H. J. (1975) Model Management in MIS, in *Seventeenth National AIDS*, Cincinati, OH.
30. Studer, R., Benjamins, V. R., and Fensel, D. (1998) Knowledge engineering: Principles and methods, *Data and Knowledge Engineering*, 25 1-2, 161-197.
31. van Harmelen, F., Patel-Schneider, P., and Horrocks, I. (2001) Reference description of the DAML+OIL ontology markup language, http://www.daml.org/2001/03/reference.
32. W3C (2004) W3C Recommendations: (1) Resource Description Framework (RDF): Concepts and Abstract Synatax, (2) RDF Vocabulary Description Language 1.0: RDF Schema, (3) OWL Web Ontology Language Semantics and Abstract Syntax, http://www.w3.org/2001/sw/.